

The image shows a massive grid of black symbols arranged in a repeating pattern. The symbols are organized into four main vertical columns. The first column contains groups of 'SSS' symbols. The second column contains groups of 'YYY' symbols. The third column contains groups of 'VVV' symbols. The fourth column contains groups of 'SSSSSSSSSS' symbols. Each group of symbols is separated by a small amount of space, and there are many such groups arranged in a grid-like fashion across the entire image area.

FILEID**RSE

0 2

RRRRRRRR		SSSSSSSS	EEEEEEEEE
RRRRRRRR		SSSSSSSS	EEEEEEEEE
RR	RR	SS	EE
RR	RR	SS	EE
RR	RR	SS	EE
RR	RR	SS	EE
RRRRRRRR		SSSSSS	EEEEEEEEE
RRRRRRRR		SSSSSS	EEEEEEEEE
RR	RR	SS	EE
RR	RR	SS	EE
RR	RR	SS	EE
RR	RR	SS	EE
RR	RR	SSSSSSSS	EEEEEEEEE
RR	RR	SSSSSSSS	EEEEEEEEE

LL			SSSSSSSS
LL			SSSSSSSS
LL			SS
LLLLLLLL		SSSSSSSS	SSSSSSSS

RSE
V04

(1)	60	HISTORY : DETAILED
(1)	78	DECLARATIONS
(1)	137	SCH\$RSE - REPORT SYSTEM EVENT
(1)	270	SCH\$UNWAIT - DECREMENT COUNT IN WAIT QUEUE
(1)	323	SITUATIONAL PRIORITY INCREMENT TABLE
(1)	342	SCH\$CHSE - CHANGE STATE TO EXECUTABLE
(1)	439	SWPO - SWAP OUT SIMPLE NON-EXECUTABLE
(1)	457	SCH\$QEND - QUANTUM END ROUTINE
(1)	612	SENDAST - Send AST to process
(1)	659	SCH\$WAKE - WAKE PROCESS INTERNAL
(1)	697	SCH\$SWPWAKE - WAKE SWAPPER PROCESS

0000 1 .TITLE RSE - REPORT SYSTEM EVENT
0000 2 .IDENT 'V04-000'
0000 3 .
0000 4 .
0000 5 .*****
0000 6 .*: COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 7 .*: DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 8 .*: ALL RIGHTS RESERVED.
0000 9 .
0000 10 .
0000 11 .*: THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12 .*: ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13 .*: INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14 .*: COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15 .*: OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16 .*: TRANSFERRED.
0000 17 .
0000 18 .*: THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19 .*: AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20 .*: CORPORATION.
0000 21 .
0000 22 .*: DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23 .*: SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24 .
0000 25 .
0000 26 .*****
0000 27 .
0000 28 .++
0000 29 .: FACILITY: EXECUTIVE, SCHEDULER
0000 30 .:
0000 31 .: ABSTRACT:
0000 32 .: THIS MODULE CONTAINS THE SYSTEM EVENT REPORTING ROUTINES AND
0000 33 .: THEIR SUPPORTING SUBROUTINES.
0000 34 .:
0000 35 .: ENVIRONMENT:
0000 36 .: MODE = KERNEL
0000 37 .--
0000 38 .:

0000 40 .SBTTL HISTORY ; DETAILED
0000 41 :
0000 42 : AUTHOR: R. HUSTVEDT CREATION DATE: 6-SEP-76
0000 43 :
0000 44 : V03-008 SSA0015 Stan Amway 8-Mar-1984
0000 45 : Allow expansion of working set if PFRATH exceeded and
0000 46 : number of active pages (PPG + GPG) exceeds 75% of WSSIZE.
0000 47 : Previously, expansion was done only if the number of active
0000 48 : pages was equal to WSSIZE.
0000 49 : (Acknowledgements go to Wayne Cardoza and Larry Kenah,
0000 50 : who both collaborated on this change.)
0000 51 :
0000 52 : V03-007 WMC0002 Wayne Cardoza 28-Feb-1984
0000 53 : Fix checks for waking swapper.
0000 54 :
0000 55 : V03-006 LY0084 Larry Yetto 10-FEB-1984 10:29
0000 56 : Fix truncation errors
0000 57 :
0000 58 : V03-005 TMK0002 Todd M. Katz 27-Dec-1983
0000 59 : Fix broken branches.
0000 60 :
0000 61 : V03-004 SSA0003 Stan Amway 5-Dec-1983
0000 62 : Added support for outswap scheduling changes.
0000 63 : Record event time for process unwait and quantum end
0000 64 : event.
0000 65 :
0000 66 : V03-003 TMK0001 Todd M. Katz 13-Nov-1983
0000 67 : Fix broken branches.
0000 68 :
0000 69 : V03-002 TCM0001 Trudy C. Matthews 4-Apr-1983
0000 70 : Change references to working set fields in PHD so that
0000 71 : they are used as unsigned words.
0000 72 :
0000 73 : V03-001 WMC0001 Wayne Cardoza 11-Mar-1983
0000 74 : Bad comparison against BORROWLIM.
0000 75 :
0000 76 :

```
0000 78 .SBTTL DECLARATIONS
0000 79
0000 80 : INCLUDE FILES:
0000 81 :
0000 82 :
0000 83     SACBDEF          : DEFINE AST CONTROL BLOCK
0000 84     SDYNDEF          : DEFINE STRUCTURE TYPE CODES
0000 85     SCEBDEF          : DEFINE COMMON EVENT BLOCK
0000 86     SIPLDEF          : IPL DEFINITIONS
0000 87     SPCBDEF          : PCB DEFINITIONS
0000 88     SPHDDEF          : PROCESS HEADER DEFINITIONS
0000 89     SPRDEF           : PROCESSOR REGISTER DEFS
0000 90     SPRIDEF          : PRIORITY INCREMENT CLASSES
0000 91     SSSDEF           : DEFINE STATUS CODES
0000 92     SSTATEDEF         : STATE DEFINITIONS
0000 93     SWQHDEF          : WAIT QUEUE HEADER DEFINITIONS
0000 94 :
0000 95 : MACROS:
0000 96 :
0000 97     .MACRO EVENT,EVTN,STATLIST,EACTION,CONT=0
0000 98     .IF NB,EVTN
0000 99     EVTS_`EVTN==EVTCTR
0000 100    .ENDC
0000 101    EVTCTR=EVTCTR+1
0000 102    .WORD EACTION-STACT
0000 103    RSE...=.
0000 104    .PSECT AES2,BYTE
0000 105    STMSK=CONT
0000 106    .IRP ST,<STATLIST>
0000 107    STMSK=STMSK+<1@SCHSC_`ST>
0000 108    .ENDR
0000 109    .LONG STMSK
0000 110    .PSECT AES1,BYTE
0000 111    .=RSE...
0000 112    .ENDM EVENT
0000 113
0000 114 :
0000 115 : GENERATE MASK FOR WAIT STATES
0000 116 :
0000 117 : GMASK STATENAME
0000 118 :
0000 119 :
0000 120     .MACRO GMASK,STATE
0000 121     ST=SCHSC `STATE
0000 122     WAITST=WAITST+<1@ST>
0000 123     .ENDM GMASK
0000 124 :
0000 125 : EQUATED SYMBOLS:
0000 126 :
0000 127     EVTCTR=0          : INITIALIZE EVENT COUNTER TO 0
0000 128     WAITST=0          : INITIALIZE WAIT STATE MASK
0000 129     ASTEXIT=0          : AST EXIT CHANGE MODE CODE
0000 130 :
0000 131 : OWN STORAGE:
0000 132 :
0000 133     .PSECT AES2,BYTE : STATE EVENT MASK PSECT
0000 134     STET=   .          : BASE OF STATE EVENT TABLE
```

RSE
V04-000

- REPORT SYSTEM EVENT
DECLARATIONS

I 2

16-SEP-1984 01:06:34 VAX/VMS Macro V04-00
5-SEP-1984 03:47:04 [SYS.SRC]RSE.MAR;1

Page 4
(1)

00000000 135

.PSECT AES1,BYTE

RSE
V04

0000 137 .SBTTL SCH\$RSE - REPORT SYSTEM EVENT
 0000 138
 0000 139 :++
 0000 140 : FUNCTIONAL DESCRIPTION:
 0000 141 : SCH\$RSE RECEIVES SYSTEM EVENT REPORTS FROM VARIOUS SOURCES
 0000 142 : AND PERFORMS THE APPROPRIATE ACTION FOR THE SPECIFIED PROCESS.
 0000 143 : EVENT REPORTING MUST BE PERFORMED WITH IPL=IPL\$ SYNCH.
 0000 144 : AS A SIDE EFFECT OF AN EVENT REPORT, THE RESCHEDULING INTERRUPT
 0000 145 : MAY BE TRIGGERED IF APPROPRIATE.
 0000 146 :
 0000 147 : CALLING SEQUENCE:
 0000 148 : BSB/JSB SCH\$RSE
 0000 149 : .BYTE EVTS_EVENTNAME
 0000 150 :
 0000 151 : THIS CALLING SEQUENCE IS GENERATED BY THE RPTEVT SYSTEM MACRO
 0000 152 :
 0000 153 : REPEVT EVENTNAME
 0000 154 :
 0000 155 : INPUT PARAMETERS:
 0000 156 : R2 - SITUATIONAL PRIORITY INCREMENT CLASS NUMBER
 0000 157 : R4 - PCB ADDRESS OF PROCESS FOR WHICH EVENT IS REPORTED
 0000 158 :
 0000 159 : EVENT NUMBER CONTAINED IN BYTE LOCATED BY ADDRESS AT TOP
 0000 160 : OF STACK. @SP
 0000 161 :
 0000 162 : IMPLICIT INPUTS:
 0000 163 : SCHEDULER DATA BASE
 0000 164 :
 0000 165 : OUTPUT PARAMETERS:
 0000 166 : NONE
 0000 167 :
 0000 168 : IMPLICIT OUTPUTS:
 0000 169 : NONE
 0000 170 :
 0000 171 : COMPLETION CODES:
 0000 172 : NONE
 0000 173 :
 0000 174 : SIDE EFFECTS:
 0000 175 : A RESECHEDULING INTERRUPT MAY BE REQUESTED IF THE SPECIFIED
 0000 176 : PROCESS IS HIGHER IN PRIORITY THAN THE CURRENT PROCESS.
 0000 177 :
 0000 178 :--
 0000 179 :
 0000 180 SCH\$RSE:: : REPORT SYSTEM EVENT

53 00 BE 9A	0000	MOVZBL	@SP,R3	: GET EVENT NUMBER
6E D6	0004	INCL	(SP)	: UPDATE RETURN ADDRESS
51 2C A4 3C	0006	MOVZWL	PCBSW STATE(R4),R1	: GET CURRENT STATE NUMBER
0000'CF43 DO	000A	10\$: MOVL	W\$TET[R3],R0	: GET STATE MASK FOR EVENT
06 50 51 E0	0010	BBS	R1,R0,ACTION	: DO ACTION IF STATE BIT SET
F1 50 53 D6	0014	INCL	R3	: CHECK NEXT ACTION
E8 0016	0016	BLBS	R0,10\$: IF CONTINUATION
05 0019	0019	RSB		: OTHERWISE IGNORE EVENT
0A' 00 53 CF	001A	190 ACTION: CASEL	R3,#0,S#MAXEV	: SWITCH ON EVENT NUMBER(UPDATED)
	001E	191 STACT:		: BASE OF ACTION TABLE
	001E	192 EVENT	AST,<-	: AST EVENT
	001E	193	CEF,-	: COMMON EVENT FLAG WAIT

001E	194		COLPG,-	: COLLIDED PAGE WAIT
001E	195		FPG,-	: FREE PAGE WAIT
001E	196		HIB,-	: RESIDENT HIBERNATE
001E	197		HIBO,-	: NON-RESIDENT HIBERNATE
001E	198		LEF,-	: LOCAL EVENT FLAG WAIT
001E	199		LEFO,-	: LOCAL EVENT FLAG WAIT (NON-RES)
001E	200		MWAI,-	: MUTEX WAIT
001E	201		PFW,-	: PAGE FAULT WAIT
001E	202		>,EVENTE	: AST EXECUTABLE STATE CHANGE
00000000	0020	203	EVT\$_COLPGA==EVT\$_AST	: USE SAME EVENT FOR COLLIDED PAGE AVAIL
0020	204			
0020	205			
0020	206	EVENT	EVENT,<LEF>,LEFEVT,CONT=1	: EVENT FLAG SETTING
0022	207	EVENT	,<CEF>,CEFEVT,CONT=1	: COMMON EVENT FLAG SET
0024	208	EVENT	,<LEFO>,EVENTE	: NON-RESIDENT LOCAL EVENT
0026	209			
0026	210	EVENT	FPGA,<-	: FREE PAGE AVAILABLE
0026	211		FPG,-	: FREE PAGE WAIT STATE
0026	212		>,EVENTF	: EXECUTABLE STATE CHANGE
0028	213	EVENT	WAKE,<-	: WAKE EVENT
0028	214		HIB,-	: RESIDENT HIBERNATE
0028	215		HIBO,-	: NON-RESIDENT HIBERNATE
0028	216		>,EVENTE	: EXECUTABLE
002A	217			
002A	218	EVENT	RESUME,<-	: RESUME EVENT
002A	219		SUSP,-	: RESIDENT SUSPENDED
002A	220		SUSPO,-	: NON-RESIDENT SUSPENDED
002A	221		>,EVENTE	: EXECUTABLE
002C	222			
002C	223	EVENT	PFCOM,<-	: PAGE FAULT COMPLETE EVENT
002C	224		PFW,-	: PAGE FAULT WAIT
002C	225		>,EVENTE	: EXECUTABLE
002E	226			
002E	227	EVENT	SETPRI,<-	: SET PRIORITY EVENT
002E	228		COM,-	: RESIDENT COMPUTE
002E	229		COMO,-	: NON-RESIDENT COMPUTE
002E	230		>,EVENTF	: EXECUTABLE
0030	231			
0030	232	EVENT	SWPOUT,<-	: SWAP OUT EVENT
0030	233		HIB,-	: RESIDENT HIBERNATE
0030	234		LEF,-	: RESIDENT LOCAL EVENT FLAG WAIT
0030	235		SUSP,-	: RESIDENT SUSPENDED
0030	236		>,SWPO,CONT=1	: SIMPLE SWAP OUT
0030	237	EVENT	<-	: SWAP OUT EVENT CONINUATION
0032	238		COM,-	: RESIDENT COMPUTE
0032	239		>,SWPOE	: EXECUTABLE OUTSWAP
0032	240			
0032	241			
0034	242			
0034	243			
0034	244			: IF CASE FALLS THROUGH, THEN BUGCHECK WITH ILLEGAL EVENT NUMBER.
0034	245			
0034	246			
0000000A	0034	247	MAXEVT=EVTCTR-1	: MAXIMUM EVENT NUMBER
0034	248		BUG_CHECK ILLEVNUM,FATAL	: ILLEGAL EVENT NUMBER
0038	249			

		0038	251	:	COMMON EVENT FLAG SET	
		0038	252	:		
		0038	253	:		
		0038	254	CEFEVT:	BLBC PCB\$L_STS(R4),EVENTE	: BR IF NOT RESIDENT
OE 24 A4	E9	0038	255	LEFEVT:	MOVL PCB\$L_PHD(R4),R3	: LOCAL EVENT FLAG SET FOR RESIDENT PROCESS
53 6C A4	D0	003C	256		ADDL #4,PHD\$L_PC(R3)	: POINT TO PHD
00C0 C3 04	C0	0040	258		MOVZWL #SSS_NORMAL,PHD\$L_R0(R3)	: SKIP PC OVER CHMK INSTRUCTION
0088 C3 01	3C	0045	259		; SET NORMAL COMPLETION FOR WAIT	
		004A	260	:		
		004A	261	:	EVENT EXECUTABLE ACTION ROUTINE	
		004A	262	:		
		004A	263	:		
		004A	264	EVENTE:		: EVENT EXECUTABLE STATE CHANGE
88'AF	9F	004A	265	EVENTF:		: ACTIVATE WITH NO WAIT TIME ACCOUNTING
		004D	266		PUSHAB B\$CH\$CHSE	: MAKE UNWAIT EXIT THROUGH CHSE
		004D	267	:	BRB SCH\$UNWAIT	: AND FALL INTO UNWAIT
		004D	268			

M 2

004D 270 .SBTTL SCHSUNWAIT - DECREMENT COUNT IN WAIT QUEUE

004D 271 :++

004D 272 : FUNCTIONAL DESCRIPTION:
004D 273 : SCHSUNWAIT DECREMENTS THE NUMBER OF PROCESSES IN THE WAIT
004D 274 : QUEUE SELECTED BY THE SPECIFIED PCB AND STATE VALUE.

004D 275 :
004D 276 : CALLING SEQUENCE:
004D 277 : BSB/JSB SCHSUNWAIT

004D 278 :
004D 279 :
004D 280 : INPUT PARAMETERS:
004D 281 : R1 - STATE NUMBER (PRESERVED)
004D 282 : R2 - UNUSED (PRESERVED)
004D 283 : R4 - PCB ADDRESS (PRESERVED)

004D 284 :
004D 285 : IMPLICIT INPUTS:
004D 286 : PCB LOCATED BY ADDRESS IN R4

004D 287 :
004D 288 : IMPLICIT OUTPUTS:
004D 289 : COUNT IN WAIT QUEUE HEADER IS DECREMENTED IF STATE IS A WAIT
004D 290 : STATE.

004D 291 :
004D 292 :--

004D 293 :
004D 294 : SCHSUNWAIT:: : DECREMENT PROPER WAIT COUNT
004D 295 : BBC R1,B^WAITMSK,20\$: SKIP OUT IF NOT WAIT STATE
004D 296 : CMPW #SCHSC_CEF,R1 : CHECK FOR COMMON EVENT FLAG WAIT
004D 297 : BEQL 30\$: CEF WAIT
004D 298 : MULL #WQHSC_LENGTH,R1 : COMPUTE BYTE INDEX TO WQ HDR
004D 299 : MOVAB L^SCHS\$Q_WQHDR[R1],R0 : COMPUTE ADDRESS OF WAIT Q HEADER
004D 300 : 10\$: DECW WQHSW_WQCNT(R0) : DECREMENT WAIT QUEUE COUNT
004D 301 : MOVL W^EXESGL_ABSTIM,PCBSL_WAITIME(R4) ; Record event time
004D 302 : RSB : RETURN

004D 303 :
004D 304 : 30\$: MOVZBL PCBSB_WEFC(R4),R0 : WAIT CLUSTER NUMBER
004D 305 : MOVL PCBSL_EFC\$[R4][R0],R0 : GET CLUSTER ADDRESS
004D 306 : ADDL #CEBS\$_WQFL,R0 : POINT TO WAIT QUEUE HEADER
004D 307 : BRB 10\$: GO DECREMENT WAIT COUNT

004D 308 :
004D 309 : GMASK CEF : COMMON EVENT FLAG
004D 310 : GMASK LEF : LOCAL EVENT FLAG WAIT
004D 311 : GMASK LEFO : LOCAL EVENT FLAG WAIT
004D 312 : GMASK HIB : HIBERNAT WAIT
004D 313 : GMASK HIBO : HIBERNATE WAIT
004D 314 : GMASK FPG : FREE PAGE WAIT
004D 315 : GMASK COLPG : COLLISION PAGE WAIT
004D 316 : GMASK PFW : PAGE FAULT WAIT
004D 317 : GMASK SUSP : SUSPENDED WAIT
004D 318 : GMASK SUSPO : SUSPENDED WAIT
004D 319 : GMASK MUAIT : MUTEX WAIT
004D 320 : WAITMSK:.LONG WAITST : MASK OF WAIT STATES

004D 321 :
004D 322 :
004D 323 :
004D 324 :
004D 325 :
004D 326 :
004D 327 :
004D 328 :
004D 329 :
004D 330 :
004D 331 :
004D 332 :
004D 333 :
004D 334 :
004D 335 :
004D 336 :
004D 337 :
004D 338 :
004D 339 :
004D 340 :
004D 341 :
004D 342 :
004D 343 :
004D 344 :
004D 345 :
004D 346 :
004D 347 :
004D 348 :
004D 349 :
004D 350 :
004D 351 :
004D 352 :
004D 353 :
004D 354 :
004D 355 :
004D 356 :
004D 357 :
004D 358 :
004D 359 :
004D 360 :
004D 361 :
004D 362 :
004D 363 :
004D 364 :
004D 365 :
004D 366 :
004D 367 :
004D 368 :
004D 369 :
004D 370 :
004D 371 :
004D 372 :
004D 373 :
004D 374 :
004D 375 :
004D 376 :
004D 377 :
004D 378 :
004D 379 :
004D 380 :
004D 381 :
004D 382 :
004D 383 :
004D 384 :
004D 385 :
004D 386 :
004D 387 :
004D 388 :
004D 389 :
004D 390 :
004D 391 :
004D 392 :
004D 393 :
004D 394 :
004D 395 :
004D 396 :
004D 397 :
004D 398 :
004D 399 :
004D 400 :
004D 401 :
004D 402 :
004D 403 :
004D 404 :
004D 405 :
004D 406 :
004D 407 :
004D 408 :
004D 409 :
004D 410 :
004D 411 :
004D 412 :
004D 413 :
004D 414 :
004D 415 :
004D 416 :
004D 417 :
004D 418 :
004D 419 :
004D 420 :
004D 421 :
004D 422 :
004D 423 :
004D 424 :
004D 425 :
004D 426 :
004D 427 :
004D 428 :
004D 429 :
004D 430 :
004D 431 :
004D 432 :
004D 433 :
004D 434 :
004D 435 :
004D 436 :
004D 437 :
004D 438 :
004D 439 :
004D 440 :
004D 441 :
004D 442 :
004D 443 :
004D 444 :
004D 445 :
004D 446 :
004D 447 :
004D 448 :
004D 449 :
004D 450 :
004D 451 :
004D 452 :
004D 453 :
004D 454 :
004D 455 :
004D 456 :
004D 457 :
004D 458 :
004D 459 :
004D 460 :
004D 461 :
004D 462 :
004D 463 :
004D 464 :
004D 465 :
004D 466 :
004D 467 :
004D 468 :
004D 469 :
004D 470 :
004D 471 :
004D 472 :
004D 473 :
004D 474 :
004D 475 :
004D 476 :
004D 477 :
004D 478 :
004D 479 :
004D 480 :
004D 481 :
004D 482 :
004D 483 :
004D 484 :
004D 485 :
004D 486 :
004D 487 :
004D 488 :
004D 489 :
004D 490 :
004D 491 :
004D 492 :
004D 493 :
004D 494 :
004D 495 :
004D 496 :
004D 497 :
004D 498 :
004D 499 :
004D 500 :
004D 501 :
004D 502 :
004D 503 :
004D 504 :
004D 505 :
004D 506 :
004D 507 :
004D 508 :
004D 509 :
004D 510 :
004D 511 :
004D 512 :
004D 513 :
004D 514 :
004D 515 :
004D 516 :
004D 517 :
004D 518 :
004D 519 :
004D 520 :
004D 521 :
004D 522 :
004D 523 :
004D 524 :
004D 525 :
004D 526 :
004D 527 :
004D 528 :
004D 529 :
004D 530 :
004D 531 :
004D 532 :
004D 533 :
004D 534 :
004D 535 :
004D 536 :
004D 537 :
004D 538 :
004D 539 :
004D 540 :
004D 541 :
004D 542 :
004D 543 :
004D 544 :
004D 545 :
004D 546 :
004D 547 :
004D 548 :
004D 549 :
004D 550 :
004D 551 :
004D 552 :
004D 553 :
004D 554 :
004D 555 :
004D 556 :
004D 557 :
004D 558 :
004D 559 :
004D 560 :
004D 561 :
004D 562 :
004D 563 :
004D 564 :
004D 565 :
004D 566 :
004D 567 :
004D 568 :
004D 569 :
004D 570 :
004D 571 :
004D 572 :
004D 573 :
004D 574 :
004D 575 :
004D 576 :
004D 577 :
004D 578 :
004D 579 :
004D 580 :
004D 581 :
004D 582 :
004D 583 :
004D 584 :
004D 585 :
004D 586 :
004D 587 :
004D 588 :
004D 589 :
004D 590 :
004D 591 :
004D 592 :
004D 593 :
004D 594 :
004D 595 :
004D 596 :
004D 597 :
004D 598 :
004D 599 :
004D 600 :
004D 601 :
004D 602 :
004D 603 :
004D 604 :
004D 605 :
004D 606 :
004D 607 :
004D 608 :
004D 609 :
004D 610 :
004D 611 :
004D 612 :
004D 613 :
004D 614 :
004D 615 :
004D 616 :
004D 617 :
004D 618 :
004D 619 :
004D 620 :
004D 621 :
004D 622 :
004D 623 :
004D 624 :
004D 625 :
004D 626 :
004D 627 :
004D 628 :
004D 629 :
004D 630 :
004D 631 :
004D 632 :
004D 633 :
004D 634 :
004D 635 :
004D 636 :
004D 637 :
004D 638 :
004D 639 :
004D 640 :
004D 641 :
004D 642 :
004D 643 :
004D 644 :
004D 645 :
004D 646 :
004D 647 :
004D 648 :
004D 649 :
004D 650 :
004D 651 :
004D 652 :
004D 653 :
004D 654 :
004D 655 :
004D 656 :
004D 657 :
004D 658 :
004D 659 :
004D 660 :
004D 661 :
004D 662 :
004D 663 :
004D 664 :
004D 665 :
004D 666 :
004D 667 :
004D 668 :
004D 669 :
004D 670 :
004D 671 :
004D 672 :
004D 673 :
004D 674 :
004D 675 :
004D 676 :
004D 677 :
004D 678 :
004D 679 :
004D 680 :
004D 681 :
004D 682 :
004D 683 :
004D 684 :
004D 685 :
004D 686 :
004D 687 :
004D 688 :
004D 689 :
004D 690 :
004D 691 :
004D 692 :
004D 693 :
004D 694 :
004D 695 :
004D 696 :
004D 697 :
004D 698 :
004D 699 :
004D 700 :
004D 701 :
004D 702 :
004D 703 :
004D 704 :
004D 705 :
004D 706 :
004D 707 :
004D 708 :
004D 709 :
004D 710 :
004D 711 :
004D 712 :
004D 713 :
004D 714 :
004D 715 :
004D 716 :
004D 717 :
004D 718 :
004D 719 :
004D 720 :
004D 721 :
004D 722 :
004D 723 :
004D 724 :
004D 725 :
004D 726 :
004D 727 :
004D 728 :
004D 729 :
004D 730 :
004D 731 :
004D 732 :
004D 733 :
004D 734 :
004D 735 :
004D 736 :
004D 737 :
004D 738 :
004D 739 :
004D 740 :
004D 741 :
004D 742 :
004D 743 :
004D 744 :
004D 745 :
004D 746 :
004D 747 :
004D 748 :
004D 749 :
004D 750 :
004D 751 :
004D 752 :
004D 753 :
004D 754 :
004D 755 :
004D 756 :
004D 757 :
004D 758 :
004D 759 :
004D 760 :
004D 761 :
004D 762 :
004D 763 :
004D 764 :
004D 765 :
004D 766 :
004D 767 :
004D 768 :
004D 769 :
004D 770 :
004D 771 :
004D 772 :
004D 773 :
004D 774 :
004D 775 :
004D 776 :
004D 777 :
004D 778 :
004D 779 :
004D 780 :
004D 781 :
004D 782 :
004D 783 :
004D 784 :
004D 785 :
004D 786 :
004D 787 :
004D 788 :
004D 789 :
004D 790 :
004D 791 :
004D 792 :
004D 793 :
004D 794 :
004D 795 :
004D 796 :
004D 797 :
004D 798 :
004D 799 :
004D 800 :
004D 801 :
004D 802 :
004D 803 :
004D 804 :
004D 805 :
004D 806 :
004D 807 :
004D 808 :
004D 809 :
004D 810 :
004D 811 :
004D 812 :
004D 813 :
004D 814 :
004D 815 :
004D 816 :
004D 817 :
004D 818 :
004D 819 :
004D 820 :
004D 821 :
004D 822 :
004D 823 :
004D 824 :
004D 825 :
004D 826 :
004D 827 :
004D 828 :
004D 829 :
004D 830 :
004D 831 :
004D 832 :
004D 833 :
004D 834 :
004D 835 :
004D 836 :
004D 837 :
004D 838 :
004D 839 :
004D 840 :
004D 841 :
004D 842 :
004D 843 :
004D 844 :
004D 845 :
004D 846 :
004D 847 :
004D 848 :
004D 849 :
004D 850 :
004D 851 :
004D 852 :
004D 853 :
004D 854 :
004D 855 :
00

007F 323 .SBTTL SITUATIONAL PRIORITY INCREMENT TABLE
007F 324 :
007F 325 : FIXED DATA:
007F 326 :
007F 327 : SITUATIONAL PRIORITY INCREMENT TABLE
007F 328 : (INDEXED BY PRIORITY INCREMENT CLASS)
007F 329 :
007F 330 B_PINC:
00 007F 331 .BYTE 0 : CLASS 0 - NONE
02 0080 332 .BYTE 2 : CLASS 1 - I/O COMPLETE
03 0081 333 .BYTE 3 : CLASS 2 - RESOURCE AVAIL
04 0082 334 .BYTE 4 : CLASS 3 - TERM OUTPUT COMP
06 0083 335 .BYTE 6 : CLASS 4 - TERM INPUT COMP
0084 336 :
0084 337 :
0084 338 EXESTATE: : EXECUTABLE STATE MASK
00003000 0084 339 .LONG <1@SCH\$C_COM>!<1@SCH\$C_COM0>
0088 340 :

0088 362 .SBTTL SCH\$CHSE - CHANGE STATE TO EXECUTABLE
 0088 363 ++
 0088 364 :+ FUNCTIONAL DESCRIPTION:
 0088 365 : SCH\$CHSE CHANGES THE STATE OF A PROCESS, AS REPRESENTED BY
 0088 366 : ITS PCB, TO AN EXECUTABLE STATE. THE RESCHEDULING INTERRUPT
 0088 367 : WILL BE TRIGGERED IF THE PROCESS IS RESIDENT AND HAS A PRIORITY
 0088 368 : GREATER THAN THAT OF THE CURRENTLY EXECUTING PROCESS. A
 0088 369 : PRIORITY INCREMENT CLASS NUMBER SUPPLIED AS A REGISTER CONTAINED
 0088 370 : ARGUMENT IS USED TO COMPUTE THE NEW PROCESS PRIORITY FROM ITS
 0088 371 : BASE PRIORITY.
 0088 372 : CALLING SEQUENCE:
 0088 373 : BSB/JSB SCH\$CHSE
 0088 374 : INPUT PARAMETERS:
 0088 375 : R0 - NEW PRIORITY (SCH\$CHSEP ONLY)
 0088 376 : R2 - PRIORITY INCREMENT CLASS NUMBER (SCH\$CHSF ONLY)
 0088 377 : 0 => NO INCREMENT (PAGEFAULT I/O COMPLETION)
 0088 378 : 1 => NON-TERMINAL I/O COMPLETION
 0088 379 : 2 => RESOURCE AVAILABILITY
 0088 380 : 3 => TERMINAL OUTPUT COMPLETION
 0088 381 : 4 => TERMINAL INPUT COMPLETION
 0088 382 : R4 - PCB ADDRESS
 0088 383 : IMPLICIT INPUTS:
 0088 384 : SCHSAQ_COMT - COMPUTE QUEUE HEADERS FOR COM,COMO STATES
 0088 385 : SCHSGB_PRI - CURRENT PROCESS PRIORITY.
 0088 386 : OUTPUT PARAMETERS:
 0088 387 : R2 - R2 PRIORITY INCREMENT CLASS NUMBER IF SCH\$CHSE. (PRESERVED)
 0088 388 : R3 - R3 (PRESERVED)
 0088 389 : IMPLICIT OUTPUTS:
 0088 390 : SCHSAQ_COMM - VECTOR OF COMPUTE QUEUE HEADERS.
 0088 391 : SCHSGL_COMQS - COMPUTE QUEUE SUMMARY BIT VECTOR.
 0088 392 : COMPLETION CODES:
 0088 393 : NONE
 0088 394 : SIDE EFFECTS:
 0088 395 : THE PCB SPECIFIED IS REMOVED FROM ITS PRESENT STATE QUEUE
 0088 396 : AND INSERTED IN THE APPROPRIATE COMPUTE QUEUE, COM OR COMO.
 0088 397 : AT THE PRIORITY COMPUTED FOR THE SPECIFIED SITUATION CLASS.
 0088 398 : THE SUMMARY BIT FOR THE DESTINATION STATE QUEUE IS SET TO
 0088 399 : NOTE THAT IT IS OCCUPIED.
 0088 400 : IF THE NEW PRIORITY FOR THE PROCESS IS GREATER THAN THAT OF
 0088 401 : CURRENT PROCESS AND IT IS RESIDENT, THE RESCHEDULING INTERRUPT
 0088 402 : WILL BE TRIGGERED.

SCH\$CHSE:
 CLRL R0 :CHANGE TO EXECUTABLE STATE
 SUBB3 B_PINC[R2],PCBSB_PRIB(R4),R0 :CLEAR HIGH SUM BITS FOR ADDB
 CMPB R0,PCBSB_PRI(R4) ,R0 :ADD PRIORITY INCR
 BLEQ 10\$:CHECK FOR > CURRENT PRI
 MOVB PCBSB_PRI(R4),R0 :NO
 :KEEP CURRENT PRIORITY INSTEAD

50	2F	A4	50	D6	0088	394	CLRL	R0	:CHANGE TO EXECUTABLE STATE
		F1	AF42	83	008A	395	SUBB3	B_PINC[R2],PCBSB_PRIB(R4),R0	:CLEAR HIGH SUM BITS FOR ADDB
08	A4	50	04	91	0091	396	CMPB	R0,PCBSB_PRI(R4)	:ADD PRIORITY INCR
				15	0095	397	BLEQ	10\$:CHECK FOR > CURRENT PRI
50	0B	A4	90	0097	398	MOVB	PCBSB_PRI(R4),R0	:NO :KEEP CURRENT PRIORITY INSTEAD	

10 50 91 009B 399 10\$: CMPB RO,#16 : CHECK FOR RESULT >15
 04 18 009E 400 BGEQ SCH\$CHSEP : YES USE COMPUTED VALUE
 50 2F A4 90 00A0 401 MOVB PCB\$B_PRIB(R4),RO : KEEP AT BASE IF LESS
 00A4 402
 00A4 403 :
 00A4 404 : SCH\$CHSEP - SUB-ENTRY POINT WITH PRIORITY PRECOMPUTED IN RO
 00A4 405 :
 00A4 406 :
 00A4 407 SCH\$CHSEP:: :
 51 12 DB 00A4 408 MFPR #PRS IPL,R1 : ENTRY WITH PRIO IN RO
 08 51 D1 00A7 409 CMPL R1,#IPLS_SYNCH : GET IPL
 62 19 00AA 410 BLSS BADIPL : MUST BE AT SYNCH OR GREATER
 51 64 0F 00AC 411 REMQUE (R4),R1 : NO, FATAL ERROR
 1C 12 00AF 412 BNEQ 10\$: REMOVE FROM CURREN QUEUE
 51 2C A4 3C 00B1 413 MOVZWL PCB\$W_STATE(R4),R1 : CONTINUE IF QUEUE NOTEMPTY
 13 CB AF 51 E1 00B5 414 BBC R1,EXESTATE,10\$: GET OLD STATE
 51 0B A4 9A 00BA 415 MOVZBL PCB\$B_PRI(R4),R1 : NO SUMMARY BITS
 03 2C A4 E9 00BE 416 BLBC PCB\$W_STATE(R4),58 : GET CURRENT PRI
 51 20 C0 00C2 417 ADDL #32,RT : SKIP IF RESIDENT
 00 00000000'EF 51 E5 00C5 418 5\$: ADDL R1, L^SCH\$GL_COMQS,10\$: MAKE NONRES PRIO
 0B A4 50 90 00CD 419 10\$: MOVB R0,PCBSB_PRI(R4) : CLEAR PRESENCE BIT FOR STATE
 51 0C D0 00D1 420 MOVL #SCH\$C_COM,R1 : SAVE NEW PRIO
 12 24 A4 E8 00D4 421 BLBS PCB\$L_STS(R4),20\$: ASSUME COM STATE
 51 D6 00D8 422 INCL R1 : CHECK FOR RESIDENCE
 50 20 C0 00DA 423 ADDL2 #32,RO : COMO=COM+1
 00 00000000'EF 50 E2 00DD 424 BBSS R0,L^SCH\$GL_COMQS,15\$: COMO HEADERS FOLLOW COM
 01CD 30 00E5 425 15\$: BSBW SCH\$SWP_WAKE : SET SUMMARY BIT FOR NEW QUEUE
 14. 11 00E8 426 BRB 35\$: WAKE SWAPPER
 50 00000000'EF 91 00EA 427 20\$: CMPB L^SCH\$GB_PRI,RO : COMPLETE STATE CHANGE
 03 19 00F1 428 BLSS 30\$: IS PRIO GREATER THAN CURRENT PROCESS
 00 00000000'EF 50 E2 00F6 429 SOFTINT #IPLS_SCHED : NO, DONT RESCHEDULE
 2C A4 51 B0 00FE 430 30\$: BBSS R0,L^SCH\$GL_COMQS,35\$: TRIGGER RESCHEDULE INTERRUPT
 51 00000000'EF 40 7E 0102 431 35\$: MOVW R1,PCBSW_STATE(R4) : SET SUMMARY BIT FOR NEW QUEUE
 91 64 0E 010A 432 MOVAQ L^SCH\$AQ_COMT[R0],R1 : SET NEW STATE
 05 0100 433 INSQUE (R4),0(RT)+ : COMPUTE HDR ADDR
 010E 434 RSB : INSERT IN NEW QUEUE
 010E 435 : RETURN
 0112 436 BADIPL: BUG_CHECK BADRSEIPL,FATAL : BAD IPL AT ENTRANCE TO RSE

0112 439 : .SBTTL SWPO - SWAP OUT SIMPLE NON-EXECUTABLE
0112 440 :
0112 441 : SWPO - SWAP OUT ACTION ROUTINE FOR SIMPLE NON-EXECUTABLE STATES
0112 442 :
0112 443 SWPO: :
FF38 30 0112 444 BSBW : NON-EXECUTABLE OUTSWAP
2C A4 B6 0115 445 INCW : REMOVE FROM WAIT QUEUE
51 64 OF 0118 446 REMQUE : UPDATE STATE NUMBER
10 B0 64 0E 011B 447 INSQUE : REMOVE FROM WAIT QUEUE
14 A0 B6 011F 448 INCW : INSERT AT TAIL OF QUEUE
05 0122 449 WQHSL_WQBL+WQHSC_LENGTH(R0) : NOTE COUNT IN WAIT QUEUE
0123 450 RSB : EXIT
0123 451 :
0123 452 : SWPOE - SWAP OUT EXECUTABLE ACTION ROUTINE
0123 453 :
50 0B A4 9A 0123 454 SWPOE: MOVZBL PCBSB_PRI(R4),R0 : GET PRIORITY
FF7A 31 0127 455 BRW SCHSCASEP : AND CHANGE TO COMO

3C A5 00 24 A4 03 E5
 0118 C4 00000000'EF B0
 10 0B A4 91 013E
 26 19 0142

012A 457 .SBTTL SCHSQEND - QUANTUM END ROUTINE
 012A 458
 012A 459 :++
 012A 460 :
 012A 461 : FUNCTIONAL DESCRIPTION:
 012A 462 : SCHSQEND IS CALLED BY THE TIMER WHEN THE QUANTUM FOR THE CURRENT
 012A 463 : PROCESS HAS BEEN EXHAUSTED. A NEW QUANTUM IS INITIALIZED
 012A 464 : THE PROCESS PLACED AT ITS BASE PRIORITY AND THE RESCHEDULING
 012A 465 : INTERRUPT TRIGGERED. A CHECK IS MADE FOR CPU TIME LIMIT EXPIRATION
 012A 466 : AND APPROPRIATE EXIT ASTS GENERATED WHEN THE LIMIT IS REACHED.
 012A 467 : THE AUTOMATIC WORKING SET SIZE LOGIC IS INVOKED IF ENABLED TO
 012A 468 : TRADEOFF WORKING SET SIZE AGAINST PAGEFAULT RATE.
 012A 469 :
 012A 470 : CALLING SEQUENCE:
 012A 471 : BSB/JSB SCHSQEND
 012A 472 :
 012A 473 : INPUT PARAMETERS:
 012A 474 : R4 - PCB ADDRESS OF CURRENT PROCESS
 012A 475 : R5 - PROCESS HEADER ADDRESS
 012A 476 :
 012A 477 : IMPLICIT INPUTS:
 012A 478 : PCB OF CURRENT PROCESS
 012A 479 : PROCESS HEADER OF CURRENT PROCESS
 012A 480 :
 012A 481 : IMPLICIT OUTPUTS:
 012A 482 : PHDSW_QUANT - INITIALIZED TO A NEW QUANTUM
 012A 483 : PCB_SV_INQUAN - INITIAL QUANTUM FLAG CLEARED
 012A 484 :
 012A 485 :--
 012A 486 :
 012A 487 SCHSQEND:: : QUANTUM END ROUTINE

012A 488 BBCC #PCBSV_INQUAN,PCBSL_STS(R4),10\$: CLEAR INITIAL QUAN FLAG
 012F 489 10\$: MOVW SCHSGW_QUAN,PHDSW_QOANT(R5) : SET NEW QUANTUM
 0137 490 MOVL W^EXESGL_ABSTIM,PCBSL_WAITIME(R4) : Record event time
 013E 491 CMPB PCBSB_PRI(R4),#16 : CHECK FOR REAL-TIME
 0142 492 BLSS 50\$: YES

0144 493 :
 0144 494 :
 0144 495 : CHECK FOR CPU TIME LIMIT EXPIRATION
 0144 496 :
 5C A5 D5 0144 497 TSTL PHDSL_CPULIM(R5) : IS THERE ANY LIMIT?
 22 12 0147 498 BNEQ 60\$: YES, GO CHECK IT OUT
 09 24 A6 18 E0 0149 499 40\$: BBS #PCBSV_DISAWS,PCBSL_STS(R4) 45\$; BRANCH IF ADJUSTMENT DISABLED
 53 0000'CF D0 014E 500 MOVL W^SCHSGL_WSINC,R3 : ASSUME INCREMENT
 02 13 0153 501 BEQL 45\$: BR IF NO AUTO WS ADJUSTMENT
 49 10 0155 502 BSBB WSADJUST : ELSE GO DO IT
 00000000'EF D5 0157 503 45\$: TSTL L^SCHSGL_COMOQS : IS THERE ANY INSWAP PENDING?
 08 13 0150 504 BEQL 47\$: NO
 OB A4 2F A4 90 015F 505 MOVB PCBSB_PRIB(R4),PCBSB_PRI(R4) : YES, FORCE TO BASE PRIORITY
 014E 30 0164 506 BSBW SCHSSUPWAKE : AND WAKE SWAPPER
 05 016A 507 47\$: SOFTINT #IPL8_SCHED : TRIGGER RESCHEDULING INT
 0168 508 50\$: RSB : AND RETURN
 0168 509 :
 0168 510 :
 0168 511 : A non-zero limit exists, check for processor time expiration
 0168 512 : If CPU time limit is exceeded then an additional amount of time will

016B 514 : be allowed for each access mode. An AST will be issued to cause an
 016B 515 : exit for each of the access modes. The additional time allowance will
 016B 516 : be provided for each access mode.
 016B 517 :
 016B 518 :
 50 38 A5 5C A5 C3 016B 519 60\$: SUBL3 PHDSL_CPUTIM(R5),PHDSL_CPUTIM(R5),R0 : HAS LIMIT BEEN REACHED
 D6 1F 0171 520 BLSSU 40\$; NO, CONTINUE NORMALLY
 0173 521 :
 0173 522 : CPU LIMIT HAS EXPIRED. AN AST WILL BE SENT TO NOTIFY THE PROCESS
 0173 523 :
 50 0000'CF CO 0173 524 ADDL2 W^SGN\$GL_EXTRACPU,R0 : COMPUTE TOTAL AMOUNT OF EXTRA TIME
 SC A5 50 CO 0178 525 ADDL2 R0,PHDSL_CPUTIM(R5) : GIVE EXTRA TIME FOR CLEANUP
 0110 C5 50 CO 017C 526 ADDL2 R0,PHDSL_EXTRACPU(R5) : AND RECORD AMOUNT OF EXTRA TIME
 50 60 A5 9E 0181 527 MOVAB PHDSB_CPMODE(R5),R0 : GET ADDRESS OF AST ACCESS MODE
 C1 AF 9F 0185 528 PUSHAB 40\$: SET RETURN ADDRESS
 53 20AC BF 3C 0188 529 MOVZWL #SSS_EXCPUTIM,R3 : PASS EXIT STATUS TO SENDAST
 00BD 30 018D 530 SCH\$FORCEDEXIT:::
 018D 531 BSBW SENDAST : SEND AST TO PROCESS
 0190 532 :
 0190 533 : CPU TIME EXPIRATION AST HANDLER
 0190 534 :
 00 0000 0190 535 CPUABRT: WORD 0 : NULL ENTRY MASK
 BC 0192 536 CHMK S^WASTEXIT : EXIT FROM AST ROUTINE (CLEAR AST)
 0194 537 10\$: SEXIT_S 4(AP) : EXIT TO INVOKE EXIT HANDLERS
 F4 11 019E 538 BRB 10\$: JUST IN CASE
 01A0 539 :
 01A0 540 : Adjust working set size automatically to achieve desired tradeoff
 01A0 541 : between page fault rate and working set size. There are two page
 01A0 542 : fault rate thresholds: SCH\$GL_PFRATL, the lower threshold and
 01A0 543 : SCH\$GL_PFRATH, the higher threshold. Each time SCH\$QEND is invoked,
 01A0 544 : the page fault rate is computed and compared with these thresholds.
 01A0 545 : If it is above the high threshold the working set size is increased
 01A0 546 : by SCH\$GW_WSINC and if the rate is below the lower threshold, the
 01A0 547 : working set size is decreased by SCH\$GW_WSDEC. The actual adjustment
 01A0 548 : is performed by a normal kernel mode AST.
 01A0 549 :
 01A0 550 :
 01A0 551 :
 01A0 552 : Automatic adjustment of working set size is constrained by the values:
 01A0 553 : SCH\$GW_AWSMIN and WSEXTENT per process that establish upper and lower
 01A0 554 : values for automatic working set size adjustment. Working set size
 01A0 555 : adjustment is further constrained by the process quota.
 01A0 556 :
 01A0 557 :
 01A0 558 :
 01A0 559 WSADJUST:
 50 0100 C5 C3 01A0 560 SUBL3 PHDSL_TIMREF(R5),- : AUTO-ADJUST WORKING SET SIZE
 38 A5 02 12 01A4 561 PHDSL_CPUTIM(R5),R0 : COMPUTE DELTA-T
 50 D6 01A7 562 BNEQ 10\$: BR IF NON-ZERO
 01AB 563 INCL R0 : ELSE FORCE TO ONE FOR DIVIDE
 0000'CF 50 D1 01AB 564 10\$: CMPL R0,W^SCH\$GL_AWSTIME : IS THIS A MEANINGFUL INTERVAL?
 3F 19 01B0 565 BLSS NOADJUST : NO, TRY AGAIN LATER
 00FC C5 C3 01B2 566 SUBL3 PHDSL_PFLREF(R5),- : COMPUTE DELTA-PGFLT
 51 4C A5 01B6 567 PHDSL_PAGEFLTS(R5),R1 :
 00FC C5 4C A5 DO 01B9 568 MOVL PHDSL_PAGEFLTS(R5),PHDSL_PFLREF(R5) : SAVE NEW PAGE FAULT REF
 0100 C5 38 A5 DO 01BF 570 MOVL PHDSL_CPUTIM(R5),PHDSL_TIMREF(R5) : AND SAVE CPUTIME REF

51 000003E8 8F C4 01C5 571 MULL #1000,R1 ; MULTIPLY BY SCALE FACTOR
 51 50 C6 01CC 572 DIVL R0,R1 AND COMPUTE PAGEFLTS/10SEC
 00F8 C5 D0 01CF 573 MOVL R1,PHDSL_PFLTRATE(R5) SAVE CURRENT RATE
 0000'CF 51 D1 01D4 574 CMPL R1,W^SCH\$GL_PFRATH ARE WE ABOVE HIGH THRESHOLD?
 17 18 01D9 575 BGEQ ADJUSTUP YES,
 53 0000'CF CE 01DB 576 MNEGL W^SCH\$GL_WSDEC,R3 NO GET DECREMENT VALUE
 0000'CF 51 D1 01E0 577 CMPL R1,W^SCH\$GL_PFRATL ARE WE BELOW LOW THRESHOLD?
 0A 18 01E5 578 BGEQ NOADJUST NO, IN DEAD BAND -- NOTHING TO DO
 0000'CF 36 A4 B1 01E7 579 CMPW PCB\$W_PPGCNT(R4),W^SCH\$GW_AWSMIN ; ARE WE AT LOWER WS LIMIT?
 02 18 01ED 580 BLEQU NOADJUST YES, NOTHING TO DO
 39 11 01EF 581 BRB ADJUST
 05 01F1 582 NOADJUST:
 05 01F1 583 RSB
 51 18 A5 08 A5 A3 01F2 584 ADJUSTUP:
 00000000'EF 50 50 A5 3C 01F8 585 SUBW3 PHDSW_WSLIST(R5),PHDSW_W\$QUOTA(R5),R1
 0000'CF 06 1A 0205 586 MOVZWL PHDSW_WSSIZE(R5),R0 ; ASSUME HIGH LIMIT WILL BE QUOTA
 51 16 A5 08 A5 A3 0207 587 CMPL PHDSW_WSSIZE(R5),R0 GET CURRENT WORKING SET SIZE
 51 50 B1 0200 588 BGTRU W^SCH\$GL_BORROWLIM,L^SCH\$GL_FREECNT ; ARE THERE LOTS OF FREE PAGES?
 51 36 A4 34 A4 A1 0212 589 SUBW3 10S BRANCH IF MEMORY IS AT A PREMIUM
 51 50 B1 0210 590 PHDSW_WSLIST(R5),PHDSW_W\$EXTENT(R5),R1
 51 DF 1A 0210 591 105: CMPW R0,R1 ALLOW LARGER GROWTH SIZE
 51 50 B1 0212 592 BGTRU NOADJUST ARE WE AT MAXIMUM SIZE?
 51 2C 1F 0218 593 ADDW3 PCBSW_GPGCNT(R4),PCBSW_PPGCNT(R4),R1 ; GET CURRENT PHYSICAL SIZE
 52 50 FE 8F 78 0210 594 CMPW R0,R1 Be sure that pages in use don't exceed WS
 50 52 A2 0222 595 BLSSU WSERR BRANCH IF WS SMALLER THAN PAGES IN USE
 50 51 B1 0225 596 ASHL #2,R0,R2 Compute 75% of WSSIZE as page threshold
 50 C7 1F 0228 597 SUBW2 R2,R0
 50 61 A5 9E 022A 600 CMPW R1,R0
 1D 10 022E 601 ADJUST: BLSSU NOADJUST If threshold not exceeded,
 0230 602 MOVAB PHDSB_AWSMODE(R5),R0 skip WS adjustment
 0230 603 BSBB SENDAST GET ADDRESS OF AST ACCESS MODE
 51 00000000'9F 0000 0230 604 ADJWS: WORD 0 SEND AST TO PROCESS
 61 A1 94 0232 605 MOVL #CTL\$GL_PHD,R1
 0239 606 CLRB PHDSB_AWSMODE(R1) ; GET PHD ADDRESS SO
 023C 607 SADJWSL_S 4(AP) ACCESS MODE FLAG CAN BE RESET
 04 0248 608 RET ADJUST BY PARAMETER IN AST ARGLIST
 0249 609
 0249 610 WSERR: BUG_CHECK WSSIZEERR,FATAL AND RETURN
 ; WORKING SET SIZE CALC IN ERROR

024D 612 .SBTTL SENDAST - Send AST to process

024D 613 ++

024D 614 FUNCTIONAL DESCRIPTION: SENDAST IS CALLED BY SCHSQEND TO SEND ASTS TO THE

024D 615 PROCESS THAT INVOKE FUNCTIONS UNAVAILABLE TO THE ENVIRONMENT OF SCHSQEND.

024D 616 THESE INCLUDE ADJUSTING THE WORKING SET AND EXITTING.

024D 617

024D 618 INPUT PARAMETERS:

024D 619

024D 620 R0 - ADDRESS OF ACCESS MODE FOR AST
(NEGATIVE CONTENTS PREVENT SENDING AST)

024D 621 R3 - AST PARAMETER

024D 622 R4 - PCB ADDRESS

024D 623 (SP) - AST ADDRESS

024D 624 4(SP) - RETURN ADDRESS FOR THIS SUBROUTINE

024D 625

024D 626 --

024D 627 SENDAST:

50	DD	024D	628	PUSHL	R0	: SAVE ADDRESS OF ACCESS MODE	
53	DD	024F	629	PUSHL	R3	: AND AST PARAMETER	
60	95	0251	630	TSTB	(R0)	: CHECK VALUE OF ACCESS MODE	
37	19	0253	631	BLSS	10\$: DO NOT QUEUE AST IF NEGATIVE	
51	1C	0255	632	BBS	#PCBSV_DELPEN,PCBSL_STS(R4)	: NOR IF MARKED FOR DELETE	
	FDAO.	025A	633	MOVZWL	10\$: SET SIZE REQUIRED	
	29	50	025D	634	BSBW	EXESALONONPAGED	: ALLOCATE A BLOCK
0A	A2	02	0260	635	BLBC	RO,10\$: NONE, TRY LATER
08	A2	51	0263	636	MOVB	#DYNSC_ACB,ACBSB_TYPE(R2)	: SET TYPE OF STRUCTURE
14	A2	8E	0267	637	MOVW	R1,ACBSW_SIZE(R2)	: AND SIZE OF STRUCTURE
0B	A2	00	026B	638	MOVL	(SP)+,ACBSL_ASTPRM(R2)	: AND AST PARAMETER VALUE
		BE	026F	639	MOVB	0(SP),ACBSB_RMOD(R2)	: SET ACCESS MODE FOR AST
10	A2	9E	0274	640	DEC B	0(SP)+	: INDICATE SUCCESS FOR THIS ACCESS MODE
0C	A2	60	0276	641	MOVL	(SP)+,ACBSL_AST(R2)	: SET AST ADDRESS
		A4	027A	642	MOVL	PCBSL_PID(R2),ACBSL_PID(R2)	: SET PID FOR AST
		30	BB	027F	PUSHR	#^M<R4,R5>	: SAVE REGS FOR QAST
55	52	00	0281	643	MOVL	R2,R5	: SET ADDRESS OF ACB
		D4	0284	644	CLRL	R2	: NULL PRIORITY INCREMENT
	FD77	30	0286	645	BSBW	SCHSQAST	: QUEUE AST FOR PROCESS
		30	BA	0289	POP R	#^M<R4,R5>	: RESTORE PCB,PHD ADDRESSES
		05	028B	646	RSB		: EXIT
			028C	647			
			028C	648			
			028C	649			
			028C	650			
			028C	651			
			028C	652			
			028C	653			
5E	0C	C0	028C	654	10\$: ADDL #12,SP		
		05	028F	655	RSB		
			0290	656			

: Error path if nonpaged pool allocation fails or if AST access mode is
 : negative, indicating either an AST in progress (for automatic working
 : set adjustment) or all access modes are done (for CPU time limit expiration)

: CLEAN PARAMETERS FROM STACK
 : AND EXIT

0290 658 .SBTTL SCH\$WAKE - WAKE PROCESS INTERNAL
 0290 659
 0290 660 ++
 0290 661 : FUNCTIONAL DESCRIPTION:
 0290 662 : SCH\$WAKE WAKES THE PROCESS SPECIFIED BY THE PID SUPPLIED.
 0290 663
 0290 664 : CALLING SEQUENCE:
 0290 665 : BSB/JSB SCH\$WAKE
 0290 666
 0290 667 : INPUT PARAMETERS:
 0290 668 : R1 - PID OF PROCESS TO WAKE
 0290 669
 0290 670 : OUTPUT PARAMETERS:
 0290 671 : R0 - COMPLETION STATUS CODE
 0290 672 : R4 - PCB ADDRESS OF PROCESS AWAKENED
 0290 673
 0290 674 : COMPLETION CODES:
 0290 675 : SSS_NORMAL - NORMAL SUCCESSFUL COMPLETION STATUS
 0290 676 : SSS_NONEXPR - NONEXISTENT PROCESS (INVALID PID)
 0290 677
 0290 678 : ENVIRONMENT:
 0290 679 : IPL = IPL\$_SYNCH
 0290 680
 0290 681 :--
 0290 682 : SCH\$WAKE:: : WAKE PROCESS INTERNAL
 54 54 51 3C 0290 683 : GET PROCESS INDEX (PIX)
 0000'DF44 D0 0293 684 : MOVL R1,R4
 60 A4 51 D1 0299 685 : DW\$SCH\$GL PCBVEC[R4],R4 : LOOK UP PCB ADDRESS
 10 12 029D 686 : CMPL R1,PCBSL_PID(R4) : VERIFY PID
 00 24 A4 0C E2 029F 687 : BNEQ 30\$: REPORT ERROR
 52 02 9A 02A4 688 10\$: BBSS #PCBSV_WAKEOPEN,PCBSL_STS(R4),10\$; SET WAKE PENDING
 50 01 3C 02A4 689 : MOVZBL #PRIS_RESAVL,R2 : SET PRIORITY INCREMENT CLASS
 05 02A7 690 : RPTEVT WAKE : REPORT WAKE EVENT
 50 01 3C 02AB 691 : MOVZWL #SSS_NORMAL,RO : SET SUCCESS CODE
 05 02AE 692 20\$: RSB : RETURN
 50 08E8 8F 3C 02AF 693 : 694 30\$: MOVZWL #SSS_NONEXPR,RO : SET NONEXISTENT PROCESS STATUS
 05 02B4 695 : RSB :

```

02B5      .SBTTL SCHSSWPWAKE - WAKE SWAPPER PROCESS
02B5      697
02B5      698      ++
02B5      699      : FUNCTIONAL DESCRIPTION:
02B5      700      : SCHSSWPWAKE AWAKENS THE SWAPPER PROCESS TO PERFORM SOME OPERATION.
02B5      701
02B5      702      : CALLING SEQUENCE:
02B5      703      : BSB/JSB SCHSSWPWAKE
02B5      704
02B5      705      : INPUT PARAMETERS:
02B5      706      : NONE
02B5      707
02B5      708      : OUTPUT PARAMETERS:
02B5      709      : R0-R4 PRESERVED
02B5      710
02B5      711      : SIDE EFFECTS:
02B5      712      : A WAKE EVENT IS GENERATED FOR THE SWAPPER PROCESS WHICH CAN
02B5      713      : CAUSE THE PROCESSOR TO BE RESCHEDULED.
02B5      714
02B5      715      :--:
02B5      716
02B5      717      SCHSSWPWAKE::          : WAKE SWAPPER PROCESS
02B5      718      TSTL    L^SCH$GL_COMOQS   : ANY INSWAP CANDIDATES?
02B5      719      BNEQ    10$              : YES, MUST WAKE SWAPPER THEN
02B5      720      CMPL    L^SCH$GL_MFYCNT,L^SCH$GL_MFYLIM : ARE THERE MODIFIED PAGES
02B5      721      BGEQ    10$              : YES, MUST WAKE SWAPPER THEN
02B5      722      CMPL    L^SCH$GL_FREECNT,L^SCH$GL_FREELIM : DO WE NEED FREE PAGES?
02B5      723      BLSS    10$              : YES, MUST WAKE SWAPPER THEN
02B5      724      TSTW    L^SCH$GW_DELPHDCT : ARE THERE DELETED HEADERS TO PURG
02B5      725      BNEQ    10$              : YES, MUST WAKE SWAPPER THEN
02B5      726      TSTL    W^EXE$GL_PFATIM : WAS THERE A POWER FAIL RECOVERY?
02B5      727      BEQL    20$              : BR IF NONE
02B5      728      10$:     TSTB    L^SCH$GB_SIP    : SWAPPER ALREADY BUSY?
02B5      729      BNEC    20$              : BR IF YES
02B5      730      PUSHR   #^M<R0,R1,R2,R3,R4> : SAVE R0-R4
02B5      731      MOVL    W^SCH$GL_SWPPID,R1 : GET PID OF SWAPPER
02B5      732      BSBB    SCH$WAKE : AND AWAKEN IT
02B5      733      POPR    #^M<R0,R1,R2,R3,R4> : RESTORE R0-R4
02B5      734      20$:     RSB     : AND RETURN TO CALLER
02B5      735
02B5      736      .END

```

- REPORT SYSTEM EVENT

K 3

16-SEP-1984 01:06:34 VAX/VMS Macro V04-00
5-SEP-1984 03:47:04 [SYS.SRC]RSE.MAR;1Page 19
(1)

ACBSB_RMOD	= 00000008		PCBSW_STATE	= 0000002C
ACBSB_TYPE	= 0000000A		PHDSB_AWSMODE	= 00000061
ACBSC_LENGTH	= 0000001C		PHDSB_CPU MODE	= 00000060
ACBSL_AST	= 00000010		PHDSL_CPLIM	= 0000005C
ACBSL_ASTPRM	= 00000014		PHDSL_CPUTIM	= 00000038
ACBSL_PID	= 0000000C		PHDSL_EXTRACPU	= 00000110
ACBSW_SIZE	= 00000008		PHDSL_PAGEFLTS	= 0000004C
ACTION	0000001A R 03		PHDSL_PC	= 000000C0
ADJUST	0000022A R 03		PHDSL_PFLREF	= 000000FC
ADJUSTUP	000001F2 R 03		PHDSL_PFLRATE	= 000000F8
ADJWS	00000230 R 03		PHDSL_RO	= 00000088
ASTEXIT	= 00000000		PHDSL_TIMREF	= 00000100
BADIPL	0000010E R 03		PHDSW_QUANT	= 0000003C
BUGS_BADRSEIPL	***** X 03		PHDSW_WSEXTENT	= 00000016
BUGS_ILLEVNUM	***** X 03		PHDSW_WSLIST	= 00000008
BUGS_WSSIZEERR	***** X 03		PHDSW_WSQUOTA	= 00000018
B_PINC	0000007F R 03		PHDSW_WSSIZE	= 00000050
CEBSL_WQFL	= 00000014		PRS_IPL	= 00000012
CEFEVT	00000038 R 03		PRS_SIRR	= 00000014
CPUABRT	00000190 R 03		PRIS_RESAVL	= 00000002
CTL\$GL_PHD	***** X 03		RSE	= 00000034 R 03
DYNSC_ACB	= 00000002		SCH\$AQ_COMT	***** X 03
EVENTE	0000004A R 03		SCHSAQ_WQHDR	***** X 03
EVENTF	0000004A R 03		SCH\$CHSE	00000088 RG 03
EVTS_AST	= 00000000 G		SCH\$CHSEP	000000A4 RG 03
EVTS_COLPGA	= 00000000 G		SCH\$C_CEF	= 00000003
EVTS_EVENT	= 00000001 G		SCH\$C_COLPG	= 00000001
EVTS_FPGA	= 00000004 G		SCH\$C_COM	= 0000000C
EVTS_PFCOM	= 00000007 G		SCH\$C_COMO	= 0000000D
EVTS_RESUME	= 00000006 G		SCH\$C_FPG	= 0000000B
EVTS_SETPRI	= 00000008 G		SCH\$C_HIB	= 00000007
EVTS_SWPOUT	= 00000009 G		SCH\$C_HIBO	= 00000008
EVTS_WAKE	= 00000005 G		SCH\$C_LEF	= 00000005
EVTCTR	= 0000000B		SCH\$C_LEFO	= 00000006
EXESALONONPAGED	***** X 03		SCH\$C_MWAIT	= 00000002
EXESGL_ABSTIM	***** X 03		SCH\$C_PFW	= 00000004
EXESGL_PFACTIM	***** X 03		SCH\$C_SUSP	= 00000009
EXESTATE	00000084 R 03		SCH\$C_SUSPO	= 0000000A
IPLS_SCHED	= 00000003		SCH\$FORCEDEXIT	0000018D RG 03
IPLS_SYNCH	= 00000008		SCH\$GB_PRI	***** X 03
LEFEVT	0000003C R 03		SCH\$GB_SIP	***** X 03
MAXEVT	= 0000000A		SCH\$GL_AWSTIME	***** X 03
NOADJUST	000001F1 R 03		SCH\$GL_BORROWLIM	***** X 03
PCBSB_PRI	= 0000000B		SCH\$GL_COMOQS	***** X 03
PCBSB_PRIB	= 00000002F		SCH\$GL_COMQS	***** X 03
PCBSB_WEFC	= 00000002E		SCH\$GL_FREECNT	***** X 03
PCBSL_EFCS	= 00000050		SCH\$GL_FREELIM	***** X 03
PCBSL_PHD	= 0000006C		SCH\$GL_MFYCNT	***** X 03
PCBSL_PID	= 00000060		SCH\$GL_MFYLIM	***** X 03
PCBSL_STS	= 00000024		SCH\$GL_PCBVEC	***** X 03
PCBSL_WAITIME	= 00000118		SCH\$GL_PFRATH	***** X 03
PCBSV_DELPEN	= 00000001		SCH\$GL_PFRATL	***** X 03
PCBSV_DISAWS	= 00000018		SCH\$GL_SWPPID	***** X 03
PCBSV_INQUAN	= 00000003		SCH\$GL_WSDEC	***** X 03
PCBSV_WAKEPEN	= 0000000C		SCH\$GL_WSINC	***** X 03
PCBSW_GPGCNT	= 00000034		SCH\$GW_AWSMIN	***** X 03
PCBSW_PPGCNT	= 00000036		SCH\$GW_DELPHDCT	***** X 03

RSE
Symbol table

- REPORT SYSTEM EVENT

L 3

16-SEP-1984 01:06:34 VAX/VMS Macro V04-00
5-SEP-1984 03:47:04 [SYS.SRC]RSE.MAR;1

Page 20
(1)

SCHSGW QUAN
SCHSQAST
SCHSQEND
SCHSRSE
SCHSSWPWAKE
SCHSUNWAIT
SCHSWAKE
SENDAST
SGNSGL_EXTRACPU
SSS_EXCPUTIM
SSS_NONEXPR
SSS_NORMAL
ST
STACT
STET
STMSK
SWPO
SWPOE
SYSSADJWSL
SYSSEXIT
WAITMSK
WAITST
WQHSC_LENGTH
WQHSL_WQBL
WQHSW_WQCNT
WSADJUST
WSERR

***** X 03
***** X 03
0000012A RG 03
00000000 RG 03
000002B5 RG 03
0000004D RG 03
00000290 RG 03
0000024D R 03
***** X 03
= 000020AC
= 000008E8
= 00000001
= 00000002
0000001E R 03
= 00000000 R 02
= 00001000
00000112 R 03
00000123 R 03
***** GX 03
***** GX 03
0000007B R 03
= 0000FFE
= 0000000C
= 00000004
= 00000008
000001A0 R 03
00000249 R 03

+-----+
! Psect synopsis !
+-----+

PSECT name

PSECT name	Allocation	PSECT No.	Attributes	CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE
: ABS .	00000000 (0.)	00 (0.)	NOPIC USR	CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE
\$ABSS	00000000 (0.)	01 (1.)	NOPIC USR	CON	ABS	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE
AES2	0000002C (44.)	02 (2.)	NOPIC USR	CON	REL	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE
AES1	000002F9 (761.)	03 (3.)	NOPIC USR	CON	REL	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE

+-----+
! Performance indicators !
+-----+

Phase

Phase	Page faults	CPU Time	Elapsed Time
Initialization	36	00:00:00.09	00:00:01.96
Command processing	123	00:00:00.48	00:00:05.87
Pass 1	331	00:00:10.62	00:00:34.14
Symbol table sort	0	00:00:01.68	00:00:04.35
Pass 2	140	00:00:02.57	00:00:09.68
Symbol table output	18	00:00:00.13	00:00:00.13
Psect synopsis output	2	00:00:00.02	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	652	00:00:15.60	00:00:56.16

The working set limit was 1650 pages.

62765 bytes (123 pages) of virtual memory were used to buffer the intermediate code.

RSE
VAX-11 Macro Run Statistics

- REPORT SYSTEM EVENT

M 3

16-SEP-1984 01:06:34 VAX/VMS Macro V04-00
5-SEP-1984 03:47:04 [SYS.SRC]RSE.MAR;1

Page 21
(1)

**

There were 60 pages of symbol table space allocated to hold 1060 non-local and 26 local symbols.
736 source lines were read in Pass 1, producing 19 object records in Pass 2.
26 pages of virtual memory were used to define 25 macros.

+-----+
! Macro library statistics !
+-----+

Macro library name

Macros defined

\$255\$DUA28:[SYS.OBJ]LIB.MLB;1
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2
TOTALS (all libraries)

12
8
20

1119 GETS were required to define 20 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:RSE/OBJ=OBJ\$:RSE MSRC\$:RSE/UPDATE=(ENH\$:RSE)+EXECML\$/LIB

0380 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

